CLAIMS

We Claim:

1.   An automated method of establishing a filesystem comprising:

5   a) establishing a first filesystem which interfaces with devices by loading software, including a first set of drivers, into memory and initializing said first set of drivers with said devices, said first filesystem mounted on a root directory comprising a single storage device;

b) allowing input/output functionality within said first filesystem; and

10   c) while input/output functionality is made available to said first filesystem, accessing said single storage device to obtain software, including a second set of drivers, and loading said software into said memory said initializing said second set of drivers with said devices to establish a second filesystem, wherein said second filesystem is mounted on a root directory

15   comprising said single storage device and another storage device and wherein said first filesystem is rendered inactive.

2.   A method as described in Claim 1 wherein said single storage device and said another storage device comprise a mirrored pair of storage

20   devices.

3.   A method as described in Claim 1 wherein said second filesystem comprises a volume manager.

4.   A method as described in Claim 2 wherein said second filesystem comprises a volume manager which directs input/output to and from said mirrored pair of storage devices.

5   5.   A method as described in Claim 1 wherein said devices comprise: storage devices; bus controller devices; and bus devices.

6.   A method as described in Claim 1 further comprising an application program interfacing with said first filesystem and wherein said 10   second filesystem is established transparently to said application program and further comprising said application program interfacing with said second filesystem after the establishment thereof.

7.   A method as described in Claim 1 wherein said software of said c) 15   is determined dynamically.

8.   A data storage system comprising a processor, a plurality of devices and a memory wherein said memory comprises instructions for implementing a method of transparently remounting filesystems comprising:

20   a) establishing a first filesystem which interfaces with said plurality of devices by loading software, including a first set of drivers, into said memory and initializing said first set of drivers with said plurality of devices, said first filesystem mounted on a root directory comprising a single storage device;

b) activating input/output functionality within said first filesystem; and

c) while input/output functionality is made available to said first

filesystem, accessing said single storage device to obtain software, including a

second set of drivers, and loading said software into said memory and

initializing said second set of drivers with said plurality of devices to establish a

5   second filesystem, wherein said second filesystem is mounted on a root

directory comprising said single storage device and another storage device and

wherein said first filesystem is rendered inactive.

9.   A data storage system as described in Claim 8 wherein said single

10   storage device and said another storage device comprise a mirrored pair of

storage devices.

10.   A data storage system as described in Claim 8 wherein said

second filesystem comprises a volume manager.

15

11.   A data storage system as described in Claim 9

wherein said second filesystem comprises a volume manager which directs

input/output to and from said mirrored pair of storage devices.

20   12.   A data storage system as described in Claim 8 wherein said

plurality of devices comprise: storage devices; bus controller devices; and bus

devices.

13.   A data storage system as described in Claim 8 further comprising

25   an application program interfacing with said first filesystem and wherein said

second filesystem is established transparently to said application program and wherein said method further comprises said application program interfacing with said second filesystem after the establishment thereof.

5   14.   A data storage system as described in Claim 8 wherein said software of said c) is determined dynamically.

        15.   An automated method of establishing a filesystem for accessing a plurality of storage devices, said method comprising:

10   a) booting from a firmware program which comprises firmware drivers, wherein said booting comprises:

        a1) using said firmware drivers, accessing a first set of drivers from a first storage device; and

        a2) using said first set of drivers, establishing a first filesystem

15   mounted on a root directory comprising said first storage device;

        b) once said first filesystem is established, utilizing enabled input/output functionality of said first filesystem to said root directory to access a second set of drivers from said first storage device; and

        c) using said second set of drivers, establishing a second filesystem

20   mounted on a root directory comprising a plurality of storage devices including said first storage device and wherein said first filesystem is rendered inactive.

        16.   A method as described in Claim 15 wherein said a2) comprises:

        loading said first set of drivers into memory;

25   disabling said input/output functionality to said first storage device;

initializing said first set of drivers and devices associated with accessing said first storage device to attach said first filesystem to said first storage device; and

resuming said input/output functionality to said first storage device.

5

17. A method as described in Claim 15 wherein said plurality of storage devices comprise a mirrored pair of storage devices.

18. A method as described in Claim 15 wherein said second filesystem comprises a volume manager.

10

19. A method as described in Claim 15 wherein said first set of drivers are defined according to a predetermined forceload list within said firmware program.

15

20. A method as described in Claim 17 wherein said second filesystem comprises a volume manager which directs input/output to and from said mirrored pair of storage devices.

20

21. A method as described in Claim 15 further comprising an application program interfacing with said first filesystem and wherein said second filesystem is established transparently to said application program and further comprising said application program interfacing with said second filesystem after the establishment thereof.

25

22.   A method as described in Claim 15 wherein said second set of drivers is determined dynamically.

23.   A method as described in Claim 19 wherein said second set of
5   drivers is determined dynamically.

24.   An automated method of transparently transferring from a first filesystem to a second filesystem comprising:

a) booting from a firmware program which comprises firmware drivers,
10   wherein said booting comprises:

a1) using said firmware drivers, accessing a first set of drivers from a first storage device, said first set of drivers specified within a predetermined firmware listing; and

a2) using said first set of drivers, establishing a first filesystem
15   mounted on a root directory comprising said first storage device;

b) once said first filesystem is established, utilizing enabled input/output functionality of said first filesystem to said root directory to access a second set of drivers from said first storage device and software for constructing a volume manager; and

20   c) using said second set of drivers and said software, establishing a second filesystem, comprising said volume manager, and mounted on a root directory comprising a plurality of storage devices configured as at least one mirrored pair and wherein said first filesystem is rendered inactive.

25.   A method as described in Claim 24 wherein said second set of drivers is determined dynamically.

26.   A method as described in Claim 24 wherein said volume manager

5   directs input/output to and from said at least one mirrored pair.

27.   A method as described in Claim 24 wherein said second filesystem is established transparently.

10